Betr.: Fortran IV Produktspezifikation

Änderungen gegenüber Ausgabe 1

Zu I.2: Es ist zusätzlich eine Kernspeicherversion geplant (24 Bytes Kernspeicher).

Das Verarbeiten von Codeprozeduren und vorübersetzten Programmeinheiten entfällt, da Assembler und Übersetzer Moduln des gleichen Aufbaus erzeugen.

Zu I.3: Sprachumfang: Die geforderte Anweisungsreihenfolge entfällt. Marken sind als Parameter zugelassen.

Zu I.4: Maschinenausrüstung:
Für die Kernspeicherversion sind erforderlich:

24K Bytes Kernspeicher
1 Lochstreifenleser
1 Lochstreifenlocher
1 Bedienungsfernschreiber.

Zu I.5: Der Doppelpunkt nach einer Markendefinition entfällt.

27.1.1970

ZUSE KG

Blatt - Nr. der Werksunterlage		
20	1	2
19	1	
18	1	
17	1	
16	1	
15	1	
14	1	2
13	1	
12	1	
11	1	
10	1	2
9	1	2
8	1	2
7	1	2
6	1	2
5	1	2
4	1	2
3	1	2
2	1	2
1	1	2
0/2	1	2
0/1	1	2
Ausgabe	1	2
Mitteilung		
Tag		
Name		
TR		

Bemerkungen

Blatt 0/1 - 0/2 Leitblätter

Blatt 1 - 20 Produktspezifikation für den Fortran IV der Z 43

Ausgabe - Kennzeichnung oben eintragen

Tag | Ausgabe | Mitteilung | Name | TR | Freigabe:

Produktspezifikation für den Fortran IV der Z 43

A26510-A9001-X3-0-98

ZUSE KG

Blatt 0/1

2+20 Blätter

Produktspezifikation für den

Fortran IV der Z 43

Ausgabe	Tag	Mitteilung	Name
2			EPC
1			EPC

Freigabe:

Produktspezifikation für den Fortran IV der Z 43

ZUSE KG

A26510-A9001-X3-0-98

Blatt 0/2

Blätter

EUZ

Inhaltsverzeichnis

I.1 Aufgabenstellung

In dieser Schrift werden die Funktionen des Übersetzers beschrieben, der für Quellenprogramme zulässige Sprachumfang festgelegt, die erforderliche Maschinenausrüstung aufgeführt und auch auf eine Kompatibilität zu Quellenprogrammen, die für die DVA 4004 und die Serie 300 erstellt wurden, eingegangen.

I.2 Übersetzer

Der Fortran IV Übersetzer stellt für das Betriebssystem der Zuse Z 43 ein normales Anwenderprogramm dar. In einem Übersetzungslauf kann jeweils nur ein Hauptprogramm, Subroutine- oder Funktionsunterprogramm übersetzt werden. Der Übersetzer erzeugt einen Programm-Modul, der den gleichen Aufbau wie ein Assembler-Modul hat.

Für den Übersetzer sind zwei Versionen vorgesehen.

1. Kernspeicherversion (die erzeugten Informationen der einzelnen Pässe werden auf Lochstreifen ausgegeben).

2. Plattenversion.

I.3. Sprachumfang

Der Fortran IV - Sprache für die Zuse Z 43 liegt der ASA-Standard zugrunde. Es sind jedoch die folgenden Einschränkungen vorgesehen (die angegebenen Nummern beziehen sich auf den Report X3,9 - 1966):

a)	4.2.4	Typ COMPLEX entfällt
b)	7.2.1.3	COMMON/Blockname/ entfällt
c)	7.2.3.5	Skalenfaktor (nP) entfällt
d)	7.2.3.10	Eine variable Formatangabe ist nicht vorgesehen
e)	7.1.2.1.2	Assigned-GOTO entfällt
f)	7.2.1.3.1.1	Definition einer Speichereinheit. Dieses Konzept wird wie folgt modifiziert:

REAL	4 Bytes
DOUBLE PRECISION	6 Bytes
INTEGERx2	2 Bytes
INTEGER	4 Bytes
LOGICAL	2 Bytes
log. Feldelement	1 Bit

Zusätzlich zum ASA-Standard sind die folgenden Möglichkeiten vorgesehen:

1. Mixed Mode in arithmetischen Ausdrücken
2. Formale Marken.

I.4 Maschinenausrüstung

Der Übersetzer setzt die folgenden minimalen Ausrüstungen voraus:

1. Kernspeicherversion

   24K Bytes Kernspeicher
   1 Lochstreifenleser
   1 Lochstreifenlocher
   1 Bedienungsfernschreiber

2. Plattenversion

   32K Bytes Kernspeicher
   1 Lochstreifenleser
   1 Bedienungsfernschreiber
   1 Magnetplatte

Die Ausstattung der Zielmaschine - auf der das Objektprogramm ablaufen soll - kann von einer geringeren Ausbaustufe sein.

I.5 Kompatibilität

Bezüglich des Sprachumfanges wird die Kompatibilität in aufsteigender Reihenfolge angestrebt:

Fortran IV Siemens 300 alt - Fortran IV Zuse Z 43 - Fortran IV Siemens 4004 - Fortran IV Siemens 300 neu

Äußerlich unterscheiden sich die Fortran IV Programme für die Lochstreifenversion der Z 43 gegenüber einer üblichen Lochkartenversion in 2 Punkten:

a) Jede Anweisung muß mit dem Zeichen ; abgeschlossen werden,

b) Kommentare werden in Apostrophs eingeklammert ('...'), innerhalb des Kommentars selbst darf der Apostroph nicht auftreten.

II. Arbeitsweise des Übersetzers

II.1 Struktur

Der Übersetzer besteht aus 4 Pässen:

PASS 1: Syntaxcheck und Anweisungsanalyse
PASS 2: Verarbeitung des Vereinbarungsteils und Speicherplatzzuordnung
PASS 3: Erzeugen des Objektcodes in relativer Form
PASS 4: Entrelativierung und Erzeugen des Moduls.

Ein Optimierungslauf ist nicht vorgesehen. Treten in den einzelnen Pässen Fehler auf, so werden diese nach Beendigung des jeweiligen Passes ausgegeben und der Übersetzungsvorgang wird abgebrochen.

PASS 2 benötigt keine externen Informationen, da PASS 1 nur den Anweisungsteil ausgibt.

II.2 Ablauf

Der Fortran-Übersetzer wird über das Betriebssystem aufgerufen und mit Parametern versorgt. Während eines Ablaufes verarbeitet der Übersetzer nur eine bereitgestellte Quellenprogrammeinheit und liefert sie in Modulform ab.

III. Binder

Der Binder verarbeitet vom Fortran-Übersetzer erstellte Moduln, vorgefertigte Fortran-Moduln (z.B. J/O-Routinen) und vom Assembler erstellte Moduln zu einem ablauffähigen Programm.

IV. Syntax der Fortran IV Fassung für die Zuse Z43

IV.1. Benutzte Metasprache

Die hier benutzte Metasprache zur Beschreibung der Syntax von Fortran IV für die Zuse Z43 stimmt in ihrer Schreibweise weitgehend mit der Backus-Notation (1) überein. Auf die Semantik der zu behandelnden Fortran-Elemente wird in diesem Kapitel nicht eingegangen. In diesem Zusammenhang sei vielmehr auf den ASA-Standart (X3.9-1966) sowie auf die vorangegangenen Bemerkungen verwiesen.

Grundsymbole der benutzten Metasprache.

::= Ergibtsymbol

| Odersymbol

Die Metavariablen sind in spitzen Klammern eingeschlossen. Links vom Ergibtsymbol steht die Definitionsvariable und rechts der sogenannte syntaktische Ausdruck. Die Definitionsvariable kann die durch den syntaktischen Ausdruck bestimmten Zeichenfolgen als "Wert" annehmen. Das Odersymbol in der Form | besagt, daß entweder die vor oder hinter diesem Symbol stehenden Größen als "Wert" zuzuordnen sind.

(1) J.W.Backus, The Syntax and Semantics of the Proposed International Algebraic Language of the Zürich ACM-GAMM Conference.

IV.2. Syntax Fortran IV Zuse Z43 in Backus Notation

1. Grundelemente

1.1 Grundsymbole

<Grundsymbol>::= <Buchstabe>|<Ziffer>|<Sonderzeichen>

<Alphazeichen>::= <Buchstabe>|<Ziffer>

<Buchstabe>::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|
Q|R|S|T|U|V|W|X|Y|Z

<Ziffer>::= 0|1|2|3|4|5|6|7|8|9

<Sonderzeichen>::=+|-|×|/|(|)|,|.|=|␣|:|;|'|

1.2 Schlüsselworte

<Vereinbarungszeichen>::= INTEGER|REAL|LOGICAL|
DOUBLE PRECISION|DIMENSION|
COMMON|EQUIVALENCE|FORMAT|
FUNCTION|SUBROUTINE|EXTERNAL|
DATA

<Steueranweisungszeichen>::= GOTO|IF|CALL|RETURN|
CONTINUE|DO|STOP|PAUSE

<Peripherieanweisungszeichen>::= READ|WRITE|PUNCH|PRINT|
BACKSPACE|REWIND|ENDFILE

<Anweisungszeichen>::=<Steueranweisungszeichen>|
<Peripherieanweisungszeichen>

<Schlüsselwort>::=<Vereinbarungszeichen>|<Anweisungs-
zeichen>|END

1.3 Operatoren

<Summationsoperator>::=+|-

<Multiplikationsoperator>::=×|/

<Exponentiationsoperator>::=××

<arithmetischer Operator>::=+|-|×|/|××

<Vergleichsoperator>::=.LT.|.LE.|.GE.|.GT.|.EQ.|.NE.

<logischer Operator>::= .OR.|.AND.|.NOT.

<leer>::=

1.4 Bezeichnung

<Bezeichnung>::= <Buchstabe>|<Bezeichnung><Alphazeichen>

1.5 Konstanten

<ganze Zahl ohne Vorzeichen>::=<Ziffer>|
<ganze Zahl ohne Vorzei-
chen><Ziffer>

<Vorzeichen>::=<leer>|+|-

<ganze Zahl>::=<Vorzeichen><ganze Zahl ohne Vorzeichen>

<Mantissenteil>::=<ganze Zahl ohne Vorzeichen>.|
.<ganze Zahl ohne Vorzeichen>|
<ganze Zahl ohne Vorzeichen>.<ganze
Zahl ohne Vorzeichen>

<Exponententeil>::= E<ganze Zahl>

<D-Exponententeil>::= D<ganze Zahl>

<Gleitkommazahl>::=<Mantissenteil>|
<Mantissenteil><Exponententeil>|
<ganze Zahl ohne Vorzeichen>
<Exponententeil>

<Gleitkommakonstante>::=<Vorzeichen><Gleitkommazahl>

<D-Gleitkommazahl>::= <Mantissenteil><D-Exponententeil>|
<ganze Zahl ohne Vorzeichen><D-Expo-
nententeil>

<D-Gleitkommakonstante>::=<Vorzeichen><D-Gleitkommazahl>

<logische Konstante>::= .TRUE.|.FALSE.

<offene Kette>::=<Grundsymbol>|
<offene Kette><Grundsymbol>

<Kette>::= '<offene Kette>'

<Hollerithkonstante>::=<ganze Zahl ohne Vorzeichen>
H<offene Kette>|<Kette>

<Konstante>::=<ganze Zahl>|<logische Konstante>|
<Gleitkommakonstante>|
<D-Gleitkommakonstante>|
<Hollerithkonstante>

2. Variable

<Variablenbezeichnung>::=<Bezeichnung>
<einfache Variable>::=<Variablenbezeichnung>

<Indexausdruck>::=<ganze Zahl ohne Vorzeichen>|
<einfache Variable>|
<einfache Variable> <Summationsoperator>
<ganze Zahl ohne Vorzeichen>|
<ganze Zahl ohne Vorzeichen>×<einfache Variable>|
<ganze Zahl ohne Vorzeichen>×<einfache Variable><Summationsoperator><ganze Zahl ohne Vorzeichen>

<Indexliste>::=<Indexausdruck>|
<Indexliste>,<Indexausdruck>

<Feldbezeichnung>::=<Bezeichnung>

<indizierte Variable>::=<Feldbezeichnung>(<Indexliste>)

<Variable>::=<einfache Variable>|
<indizierte Variable>

3. Funktionen

<Funktionsbezeichnung>::=<Bezeichnung>
<Formelfunktionsbezeichnung>::=<Bezeichnung>
<Subroutinenbezeichnung>::=<Bezeichnung>
<Prozedurbezeichnung>::=<Funktionsbezeichnung>|
<Subroutinenbezeichnung>
<Parameterliste1>::=<Ausdruck>|<Parameterliste1>,
<Ausdruck>

```
<aktueller Parameter>::=<Ausdruck>|
                        <Feldbezeichnung>|
                        <Prozedurbezeichnung>

<aktuelle Parameterliste>::=<aktueller Parameter>|
                            <aktuelle Parameterliste>,<ak-
                             tueller Parameter>

<Funktion>::=<Funktionsbezeichnung>(<aktuelle Parameter-
              liste>)|< Formelfunktionsbezeichnung>(< Para-
              meterliste1>)
```

4. Ausdrücke

4.1 Arithmetische Ausdrücke

```
<Elementarausdruck>::=<ganze Zahl ohne Vorzeichen>|
                      <Gleitkommazahl> |
                      <D-Gleitkommazahl>|
                      <Variable>|
                      <Funktion>|
                     (<arithmetischer Ausdruck>)

<Faktor>::=<Elementarausdruck>|
           <Faktor>××<Elementarausdruck>

<Term>::=<Faktor>|
         <Term><Multiplikationsoperator><Faktor>

<arithmetischer Ausdruck>::=<Term>|
                            <Summationsoperator><Term>|
   <arithmetischer Ausdruck><Summationsoperator><Term>
```

4.2 Logische Ausdrücke

```
<Vergleich>::=<arithmetischer Ausdruck><Vergleichoperator>
              <arithmetischer Ausdruck>

<logischer Elementarausdruck>::=<Logische Konstante>|
                                <Variable>|
                                <Funktion>|
                                <Vergleich>|
                               (<logischer Ausdruck>)
```

<logischer Faktor>::=<logischer Elementarausdruck>|
.NOT.<logischer Elementarausdruck>

<logischer Term>::=<logischer Faktor>|
<logischer Term>.AND.<logischer Faktor>

<logischer Ausdruck>::=<logischer Term>|
<logischer Ausdruck>.OR.
<logischer Term>

<Ausdruck>::=<arithmetischer Ausdruck>|
<logischer Ausdruck>

5. Marken

<Marke>::=<ganze Zahl ohne Vorzeichen>

6. Anweisungen

6.1 Ergibtanweisung

<Ergibtanweisung>::=<Variable>=<Ausdruck>;|
<Funktionsbezeichnung>=<Ausdruck>;

6.2. Steueranweisungen

<Steueranweisung>::=<Sprunganweisung>|
<Bedingungsanweisung>|
<Laufanweisung>|
<Hilfsanweisung>|
<Prozeduranweisung>|
<Rücksprunganweisung>

6.2.1 Sprunganweisungen

<unbedingte Sprunganweisung>::= GOTO<Marke>;

<Markenliste>::=<Marke>|
<Markenliste>,<Marke>

<Verteileranweisung>::= GOTO(<Markenliste>),<einfache Variable>;

<Sprunganweisung>::=<unbedingte Sprunganweisung>|
<Verteileranweisung>

6.2.2 Bedingungsanweisungen

<Bedingungsanweisung>::= IF (<logischer Ausdruck>)
<Anweisung>|IF(arithmetischer Ausdruck)<Marke>,<Marke>,<Marke>;

6.2.3 Laufanweisung

<Laufangabe>::=<Laufvariable>=<Anfangswert>,<Endwert>
<Schrittangabe>

<Laufanweisung>::= DO<Marke><Laufangabe>;

<Laufvariable>::=<einfache Variable>

<Lauflistenelement>::=<ganze Zahl ohne Vorzeichen>|
<einfache Variable>

<Anfangswert>::=<Lauflistenelement>

<Endwert>::=<Lauflistenelement>

<Schrittangabe>::=<leer>|
,<Lauflistenelement>

6.2.4 Hilfsanweisungen

<Scheinanweisung>::= CONTINUE;

<Pausenanweisung>::= PAUSE;|PAUSE<Kette>;

<Stopanweisung>::= STOP;

<Hilfsanweisung>::=<Scheinanweisung>|
<Pausenanweisung>|
<Stopanweisung >

6.2.5 Prozeduranweisung

<aktuelle Call-Liste>::=<aktueller Call-Parameter>|
<aktuelle Call-Liste>,< aktueller Call-Parameter>

<aktueller Call-Parameter>::=<aktueller Parameter>|
<Hollerithkonstante>

<Prozeduranweisung>::= CALL<Subroutinenbezeichnung>;|
CALL<Subroutinenbezeichnung>
(<aktuelle Call-Liste>);

6.2.6 Rücksprunganweisung

<Rücksprunganweisung>::= RETURN;

6.3 Ein-Ausgabeanweisungen

```
<einfache Liste>::=<Variable>|
                   <Feldbezeichnung>|
                   <einfache Liste>,<einfache Liste>

<Liste>::=<einfache Liste>|
          (<einfache Liste>)|
          (<Liste>,<Laufangabe>)|
          <Liste>,<Liste>

<Ein-Ausgabeliste>::=<leer>|
                     <Liste>

<Ein-Ausgabeeinheit>::=<ganze Zahl ohne Vorzeichen>|
                       <einfache Variable>

<Format-Leseanweisung>::=READ(<Ein-Ausgabeeinheit>,
                         <Marke>)<Ein-Ausgabeliste>;

<Binär-Leseanweisung>::=READ(<Ein-Ausgabeeinheit>)
                        <Ein-Ausgabeliste>;

<Leseanweisung>::=<Format-Leseanweisung>|
                  <Binär-Leseanweisung >

<Format-Schreibanweisung>::=WRITE(<Ein-Ausgabeeinheit>,
                            <Marke>)<Ein-Ausgabeliste>;
<Binär-Schreibanweisung>::=WRITE(<Ein-Ausgabeeinheit>)
                           <Ein-Ausgabeliste>;

<Schreibanweisung>::=<Format-Schreibanweisung>|
                     <Binär-Schreibanweisung>

<Hilfs-Ein-Ausgabeanweisung>::=REWIND<Ein-Ausgabeeinheit>;|
                               BACKSPACE<Ein-Ausgabeein-
                               heit>;|ENDFILE<Ein-Ausgabe-
                               einheit>;
```

```
<Ein-Ausgabeanweisung>::=<Leseanweisung>|
                         <Schreibanweisung>|
                         <Hilfs-Ein-Ausgabeanweisung>

<Anweisung>::=<Ergibtanweisung>|
              <Steueranweisung>|
              <Ein-Ausgabeanweisung>
```

7. Vereinbarungen

```
<Vereinbarung>::=<Typenvereinbarung>|
                 <Speichervereinbarung>|
                 <Feldvereinbarung>|
                 <Formatvereinbarung>|
                 <lokale Datenzuordnung>|
                 <EXTERNAL-Vereinbarung>|
                 <Prozedurvereinbarung>|
                 <Programmvereinbarung>
```

7.1 Typenvereinbarungen

```
<Indexgrenze>::=<ganze Zahl ohne Vorzeichen>|
                <einfache Variable>

<Indexgrenzenliste>::=<Indexgrenze>|
                      <Indexgrenzenliste>,<Indexgrenze>

<Feldangabe>::=<Feldbezeichnung>(<Indexgrenzenliste>)

<Typlistenelement>::=<Variablenbezeichnung>|
                     <Feldbezeichnung>|
                     <Feldangabe>|
                     <Formelfunktionsbezeichnung>|
                     <Funktionsbezeichnung>

<Typliste>::=<Typlistenelement>|
             <Typliste>,< Typlistenelement>

<Typ>::= INTEGER|
         REAL|
         DOUBLEPRECISION|
         LOGICAL
```

<Typvereinbarung>::=<Typ><Typliste>;

7.2 Speichervereinbarungen

<Speicherlistenelement>::=<Variablenbezeichnung>|
<Feldbezeichnung>|
<Feldangabe>

<Speicherliste>::=<Speicherlistenelement>|
<Speicherliste>,<Speicherlistenelement>

<COMMON-Vereinbarung>::= COMMON<Speicherliste>;

<Variablenliste>::=<Variable>|
<Variablenliste>,<Variable>

<Äquivalenzliste>::=(<Variablenliste>)|
<Äquivalenzliste>,(<Variablenliste>)

<Äquivalenzvereinbarung>::= EQUIVALENCE<Äquivalenzliste>;

<Speichervereinbarung>::=<COMMON-Vereinbarung>|
<Äquivalenzvereinbarung>

7.3 Feldvereinbarung

<Feldliste>::=<Feldangabe>|
<Feldliste>,<Feldangabe>

<Feldvereinbarung>::= DIMENSION<Feldliste>;

7.4 Formatvereinbarungen

<Umkodierungszeichen>::=F|E|G|D|I|L|A|H|X

<Umkodierungszeichen für Gleitkomma>::=F|E|G|D

<Feldlänge>::=<ganze Zahl ohne Vorzeichen>

<Dezimalbruchlänge>::=<ganze Zahl ohne Vorzeichen>

<Wiederholungsfaktor>::=<leer>|
<ganze Zahl ohne Vorzeichen>

<Gleitkommavorspann>::=<Wiederholungsfaktor>

<Gleitkommanachspann>::=<Feldlänge>.<Dezimalbruchlänge>

<Gleitkommabeschreibung>::=<Gleitkommavorspann><Umkodierungszeichen für Gleitkomma><Gleitkommanachspann>

<Ganzzahlbeschreibung>::=<Wiederholungsfaktor>I<Feldlänge>

<Beschreibung logischer Werte>::=<Wiederholungsfaktor>L<Feldlänge>

<variable Hollerithbeschreibung>::=<Wiederholungsfaktor>A<Feldlänge>

<Hollerithbeschreibung>::=<Hollerithkonstante>|
<variable Hollerithbeschreibung>

<Zwischenraumbeschreibung>::=<Feldlänge>X

<Feldbeschreibung>::=<Gleitkommabeschreibung>|
<Ganzzahlbeschreibung>|
<Beschreibung logischer Werte>|
<Hollerithbeschreibung>|
<Zwischenraumbeschreibung>

<Satztrenner>::=/|<Satztrenner>/

<Feldtrenner>::=,|<Satztrenner>

<Feldgruppe>::=<Feldbeschreibung>|
<Feldgruppe><Feldtrenner><Feldgruppe>|
<Wiederholungsfaktor>(<Feldgruppe>)

<Blockvorspann>::=<leer>|<Satztrenner>

<Blocknachspann>::=<leer>|<Satztrenner>

<Blockbeschreibung>::=<Blocknachspann>|
<Blockvorspann><Feldgruppe><Blocknachspann>

<Formatbeschreibung>::= FORMAT(<Blockbeschreibung>);

7.5 Lokale Datenzuordnung

<Konstantenliste>::=<Konstante>|
<ganze Zahl ohne Vorzeichen><Konstante><Konstantenliste>,<Konstantenliste>

<Datenliste>::=<Variablenliste>/<Konstantenliste>/|
<Datenliste>,<Datenliste>

<lokale Datenzuordnung>::= DATA<Datenliste>;

7.6 EXTERNAL-Vereinbarung

<EXTERNAL-Liste>::=<Prozedurbezeichnung>|
<EXTERNAL-Liste>,<Prozedurbezeichnung>

<EXTERNAL-Vereinbarung>::= EXTERNAL<EXTERNAL-Liste>;

7.7 Prozedurvereinbarung

<Prozedurvereinbarung>::=<Formelfunktionsvereinbarung>|
<Funktionsvereinbarung>|
<Subroutinenvereinbarung>

7.7.1 Formelfunktionsvereinbarung

<formale Formelfunktionsparamerliste>::=<Variablenbezeichnung>|<formale Formelfunktionsparameterliste>,<Variablenbezeichnung>

<Formelfunktionsvereinbarung>::=<Formelfunktionsbezeichnung>(<formale Formelfunktionsparameterliste>)=<Ausdruck>;

7.7.2 Funktionsvereinbarung

<formaler Parameter>::=<Variablenbezeichnung>|
<Feldbezeichnung>|
<Prozedurbezeichnung>

<formale Parameterliste>::=<formaler Parameter>|
<formale Parameterliste>,<formaler Parameter>

```
<Vereinbarungselement>::=<Typvereinbarung>|
                         <Speichervereinbarung>|
                         <Feldvereinbarung>|
                         <Marke>:<Formatvereinbarung>|
                         <EXTERNAL-Vereinbarung>|
                         <lokale Datenzuordnung>|
                         <Formelfunktionsvereinbarung>

<Vereinbarungsteil>::=<leer>|
                      <Vereinbarungsteil><Vereinbarungs-
                       element>

<Anweisungselement>::=<Anweisung>|
                      <Marke> <Anweisung>|
                      <Marke> <Formatvereinbarung>

<Anweisungsteil>::=<Anweisungselement>|
                   <Anweisungsteil><Anweisungselement>

<Prozedurrumpf>::=<Vereinbarungsteil><Anweisungsteil>END;

<Typus>::=<leer>|<Typ>

<Funktionsvereinbarung>::=<Typus>FUNCTION<Funktions-
                          bezeichnung>(<formale Para-
                          meterliste>);<Prozedurrumpf>
```

7.7.3 Subroutinenvereinbarung

```
<formaler Parameterteil>::=<leer>|
                           (<formale Parameterliste>)

<Subroutinenvereinbarung>::= SUBROUTINE<Subroutinen-
        bezeichnung><formaler Parameterteil>;<Prozedurrumpf>
```

7.8 Programmvereinbarung

```
<Hauptprogramm>::=<Vereinbarungsteil><Anweisungsteil>END;

<Unterprogramm>::=<Funktionsvereinbarung>|
                  <Subroutinenvereinbarung>

<Unterprogrammfolge>::=<Unterprogramm>|
                       <Unterprogrammfolge><Unterprogramm>

<ausführbares Programm>::=<Hauptprogramm>|
                  <Hauptprogramm><Unterprogrammfolge>
```